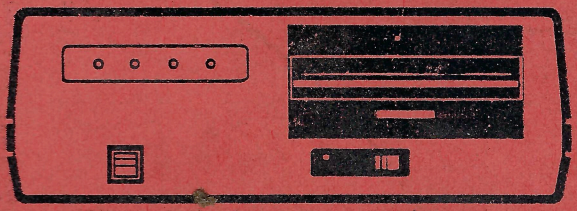
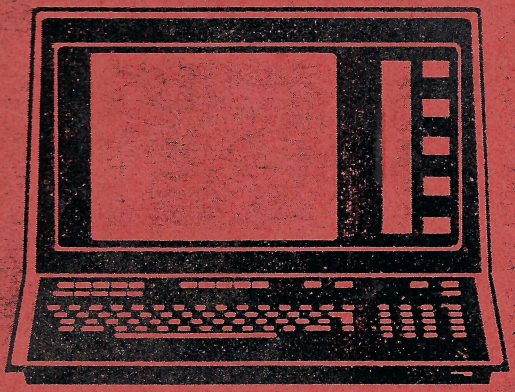
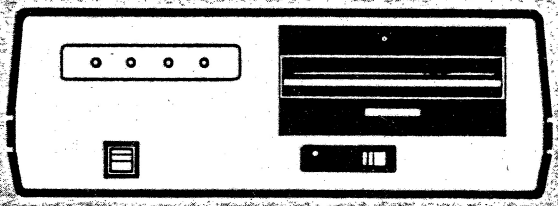
4907 FILE MANAGER OPERATOR'S POCKET REFERENCE

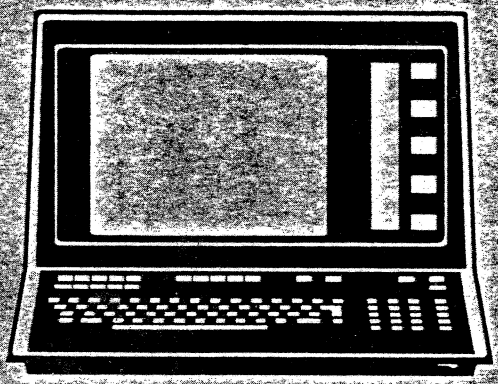




Tektronix®
COMMITTED TO EXCELLENCE

4907 FILE MANAGER OPERATOR'S POCKET REFERENCE





Tektronix®
COMMITTED TO EXCELLENCE

INTRODUCTION

This pocket reference is intended as a quick, easy-to-use reference for programming or operating the 4907 File Manager. Preparatory steps such as disc loading, as well as all other operational details, are explained in the 4907 File Manager Operator's Manual.

This pocket reference includes the following:

- GENERAL SEQUENCE FLOW CHART
- COMMAND SUMMARIES
- SAMPLE I/O PROGRAM
- HOW TO WRITE A FILE IDENTIFIER
- SAMPLE DEVICE AND FILE STATUS MESSAGES

Copyright © 1978 by Tektronix, Inc., Beaverton, Oregon. Printed in the United States of America. All rights reserved. Contents of this publication may not be reproduced in any form without permission of Tektronix, Inc. U.S.A. and Foreign TEKTRONIX products covered by U.S. and foreign patents and/or patents pending.

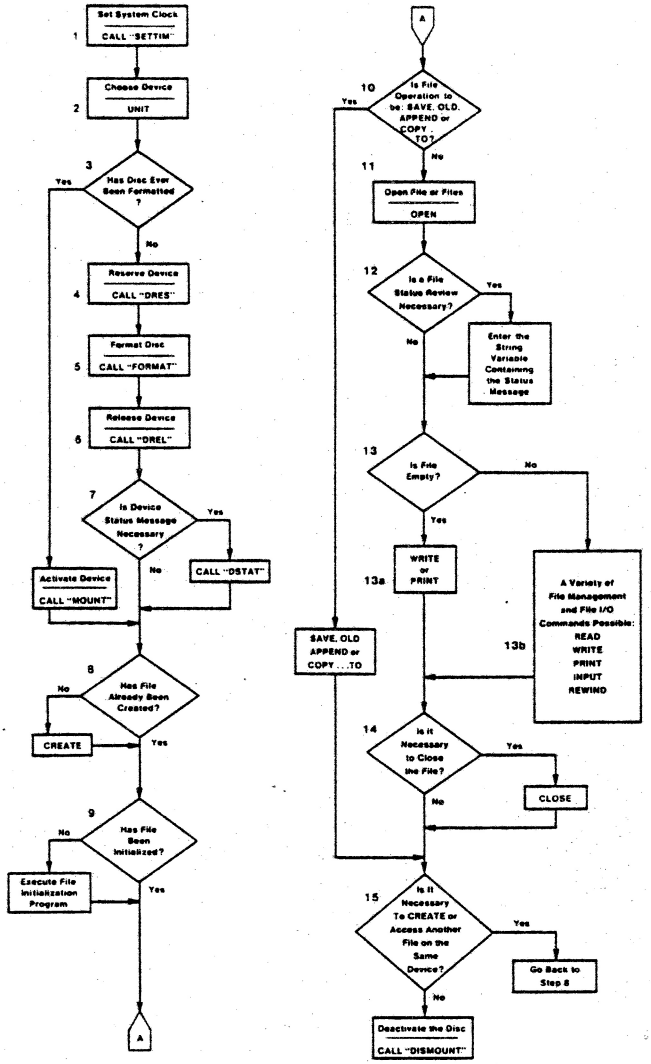
TEKTRONIX is a registered trademark of Tektronix, Inc.

070-2381-00

JAN 1978

GENERAL SEQUENCE FLOW CHART

This general sequence flow chart shows the steps necessary to create and use files. Specific command details are described in the 4907 File Manager Operator's Manual.



1. **Set System Clock**

   The CALL "SETTIM" command starts the system clock so the time and date of various system activities are recorded. This command is required each time the system is turned off and restarted. There is no "default" time provided. This command is mandatory.

2. **Choose Device**

   The UNIT command identifies which device is to be the "current" device, the one the system accesses whenever it receives a command that does not specifically name some other device. The address to be used in this command can be seen on or around the face of the device. Device 0 will always be accessed if no UNIT command is executed.

3. **Has Disc Ever Been Formatted?**

   All discs must be formatted before they are used for the first time or when all existing information is to be replaced. If the disc already has been formatted and has valid volume label information, no formatting is necessary. CALL "MOUNT" notifies the system that there is a formatted disc ready for use.

4. **Reserve Device**

   The device must be reserved with CALL "DRES" before any preparatory disc operation, such as formatting, can take place.

5. **Format Disc**

   The disc may now be formatted with the CALL "FORMAT" command. This command also executes an automatic CALL "MOUNT" to notify the system that there is a formatted disc ready for use.

6. **Release Device**

   The reserved device must now be released with a CALL "DREL" command. If the device is not released, no files can be opened.

7. **Is Device Status Message Necessary?**

   If you wish to confirm that the correct disc is in the drive, or what the specifications of that disc are, you may execute a CALL "DSTAT" command. This also allows you to see if the disc has been formatted.

At this point, preparatory work involving the disc is complete. Steps 8 through 16 show commonly used file operations on a disc prepared as shown in Steps 1 through 7.

8. Has File Already Been Created?

If you wish to store information and no file has yet been created to receive it, the CREATE command must be executed.[1] This command allows you to specify: the name of the file and its attributes, the length of the file, the number of logical records the file will contain, and whether the file is random or sequential.

9. Has File Been Initialized?

Although information may be "randomly" read from a random file, it must be entered sequentially unless the records are first filled with "blank strings." See file initializing program in the section SAMPLE I/O PROGRAM. This program enters blank strings into a file, allowing random data entry.

10. Is File Operation to be SAVE, OLD, APPEND or COPY . . . TO?

These operations may be carried out without the OPEN and CLOSE commands.

11. Open File or Files

An OPEN command must be executed for those file operations listed in Step 13. This command also generates and stores a file status message for each file opened. Another UNIT command, as in Step 2, is necessary to access a file on a different device.

12. Is a File Status Review Necessary?

It may be necessary to review the status of the file or files just opened. The parameters include the time and date the file was created, altered, and last used, as well as its name, attributes, and space specifications. To see what details are included and how they are displayed, turn to SAMPLE DEVICE AND FILE STATUS MESSAGES. If you do wish to review the file status, enter the string variable that was specified in the last field of the OPEN command.

[1] Except for SAVE which can create its own file.

13. Is File Empty?

If the file has just been created, it contains no information. Information now may be entered into the file using a WRITE or PRINT command.

If the file is full or partially full, as indicated in a file status message, a variety of file management and file I/O operations may be executed. For example, the following commands are now possible:

WRITE . . . for entering binary information
PRINT . . . for entering ASCII information

READ . . . for retrieving binary information
INPUT . . . for retrieving ASCII information

CALL "REWIND" resets the file pointer to the beginning of the file.

These commands are used to carry out some of the more common file management operations. The command examples in How to Write a Command, in the 4907 File Manager Operator's Manual, outline other operations that are less common.

14. Is It Necessary to Close the File?

If no further file access is necessary, the file may be closed by executing a CLOSE command. It is not necessary to close the file in order to open another, because up to nine files may be opened and used simultaneously.

15. Is It Necessary to CREATE or Access Another File on the Same Device?

Your operation may require placing data into or retrieving data from more than one file at a time. If you wish to access or create another file on the same device, go back to Step 8. If no further access to this device is necessary, it may be dismounted by executing CALL "DISMOUNT". All files must be closed before executing this command. If you wish to access a file on a different device, go back to Step 2.

COMMAND SUMMARIES

Information in parentheses helps describe the preceding field (except in TYPE and ON . . . EOF summaries).

APPEND F.I. [,"ASCII"]; target line number, [, increment between line numbers]

Adds the program in specified file to memory at the stated line number.

ASSIGN F.I.;attributes (R,U,S,C,N,M)

Changes file attributes such as:
R Private
U Public
S Scattered
C Contiguous
N Not Compressible
M Compressible

CLOSE [lfn] (logical file number)

Closes one or all files.

CALL "COMPRS", device address, compress control (1 or 0)

Collects all scattered free space on the disc into one contiguous space.

COPY . . . F.I. #1, source device address TO F.I. #2, target device address

Copies files to the same or another disc.

CREATE F.I. [,attributes]; number of logical records, record length

Defines a new empty file.

CALL "CUSTAT", target string variable

Generates a device status message for each device interfaced to the 4907 File Manager controller and stores them in the specified string variable.

DELALL

Erases everything currently in 4051 memory and closes all files. See Command Descriptions in the 4907 File Manager Operator's Manual.

DIRECTORY [I/O address:] [format code (0, 1, or 2) [,F.I.]]

Lists (to 4051 screen or mag tape) facts about specified files in storage.

Depending on the format code entry, listings can include:

0 File names only.
1 File names and dates.
2 Full file status.

CALL "DISMOUNT", device address

Deactivates specified device when all open files are closed. Device remains active until all files are closed. Until files are closed, I/O may continue but no new file can be opened.

CALL "DREL", device address

Releases a reserved disc drive.

CALL "DRES", device address

Reserves a disc drive.

CALL "DSTAT", device address, target string variable

Generates a status message about the disc in the drive or the device specified.

CALL "DUP", source device address, target device address, compress control (0 or 1)

Duplicates contents from one disc to another.

END

Stops program execution and closes files.

CALL "FFRMT", device address, volume I.D., 1, 1, owner I.D., master password, 1st, 2nd, 3rd, 4th, and 5th level chains

Reformats a disc without searching first for defective sectors. See CALL "FORMAT".

CALL "FILE", device address, F.I., target string variable

Stores status message of named file in specified string variable.

CALL "FMVALS", target numeric variable, target string variable

Sends the number of the current device to the numeric variable specified and the name of the current library to the string variable specified.

CALL "FORMAT", device address, volume I.D., 1, 1, owner I.D., master password, 1st, 2nd, 3rd, 4th, and 5th level chains

Prepares a disc for data storage. Records an information label on a new disc after searching for bad sectors. The disc is then automatically mounted. Any previous data is now erased.

CALL "HERRS", device address, number of retries in last I/O, no. of accumulated retries, no. of successful I/O recoveries, no. of unsuccessful I/O operations

Generates a count of I/O errors. See Command Description in the 4907 File Manager Operator's Manual or 4907 Service Manual.

INIT

Resets all 4051 string, numeric,,and array variables to an undefined state and closes all files.

INPUT #lfn (logical file number) [,record number]; target variables [,target variables] . . .

Inputs ASCII file data from the disc to the 4051; sequential or random access is permitted.

KILL F.I. [,master password]

Deletes existing files.

CALL "MOUNT", device address, target string variable

Activates disc and returns disc status message.

CALL "MRKBBG", device address, volume I.D., master password, address of defective space

Marks defective blocks on the disc so they aren't used again.

CALL "NEXT", lfn (logical file number), target string variable

Closes current file and opens the next one.

OLD F.I. [,"ASCII"]

Works like 4051 keywords "OLD" and "BOLD" (see the 4051 Graphic System Reference Manual; also see Command Descriptions in the 4907 File Manager Operator's Manual.)

ON EOF (lfn) (logical file number)

See 4051 Graphic System Reference Manual.

OPEN F.I.[,"G"]; lfn (logical file number), type of access (R,F,U), target string variable

Opens a file for I/O operation.

PRINT #lfn [,record number]: data item, data item . . .

Transfers ASCII data from the 4051 to the disc or tape.

READ #lfn(logical file number) [,record number]: target variables [,target variables] . . .

Transfers binary data from the open file to the.4051.

CALL "RENAME", device address, old F.I., new F.I.

Renames files and changes passwords.

CALL "REWIND", lfn (logical file number)

Sets logical file pointer back to beginning of file.

COMMAND SUMMARIES (Continued)

SAVE F.I. [,"ASCII"]; [line number [beginning, ending line number]]
Stores program into disc file. A file will be created if none exists.

SECRET
Prevents future program listing. See Command Descriptions in the 4907 File Manager Operator's Manual.

CALL "SETTIM", date and time
Initializes the system clock.

CALL "SPACE", lfn (logical file number), requested file size, target numeric variable, target numeric variable
Changes the size of a file to the size specified.

CALL "TIME", target string variable
Stores the current time from the system clock in a specified string variable.

TYPE (lfn) (logical file number)
Works the same as TYP(0) does for internal magnetic tape. See the 4051 Graphic System Reference Manual.

UNIT device address
Selects current device address for all commands that do not contain a field for a device address.

CALL "UNIT", device address
Same as UNIT but allows a variable in the device address field.

CALL "USERLIB", partial F.I.
Specifies current library. The default current library is "SCRATCHLIB".

WRITE #lfn (logical file number) [,record number]: data item[,data item]
Transfers binary data from the 4051 to an open file.

SAMPLE I/O PROGRAM

The program listed below shows the basic steps necessary to create files and write and print to those files. It also shows how to retrieve and print the information stored in those files.

If error messages occur, check Error Messages in the 4907 File Manager Operator's Manual. Be sure that all entries are correct and that all preparatory steps described in the 4907 File Manager Operator's Manual have been performed.

A description of the program operations and a sample of the program output follow the program listing below.

```
100 INIT
110 DIM R$(2000),F$(200)
120 CALL "TIME",R$
130 IF LEN(R$)>0 THEN 170
140 PRINT "ENTER DATE AND TIME (DD-MOM-YY HH:MM:SS):";
150 INPUT A$
160 CALL "SETTIM",A$
170 PRINT "HOW MANY DEVICES ON YOUR SYSTEM?:";
180 INPUT N
190 DIM D(N)
200 PRINT "ENTER DEVICE ADDRESSES:";
210 INPUT D
220 FOR I=1 TO N
230 PRINT "JJTHIS IS A SAMPLE PROGRAM FOR DEVICE ";D(I);"┘"
240 CALL "UNIT",D(I)
250 CALL "DRES",D(I)
260 CALL "FORMAT",D(I),"SAMPLE",1,1,"OWNER","PASS",3,3,3,3,3
270 CALL "DREL",D(I)
280 CREATE "ASCFILE","A";1,0
290 CREATE "BINFILE";1,128
300 OPEN "ASCFILE";1,"F",F$
310 PRINT #1:"THIS IS AN ASCII SAMPLE (SEQUENTIAL FILE)"
320 OPEN "BINFILE";2,"F",F$
330 WRITE #2,1:"THIS IS A BINARY SAMPLE (RANDOM FILE)"
340 CALL "REWIND",1
350 INPUT #1:S$
360 PRINT S$
370 READ #2,1:S$
380 PRINT S$
390 CLOSE
400 NEXT I
410 END
```

Program Description

100 Initialize.
110
120
130
140
150 Set system clock, if necessary.
160
170
180 Enter the total devices on your system (1, 2, or 3).
190
200
210 Enter the device addresses (0 or 0, 1, etc.).
220
230 Print heading.
240 Set unit number.

250
260 } Format disc.
270

280 Create an ASCII, sequential file.
290 Create a binary, random file

300
310 } Open each file and store message.
320
330

340 Rewind sequential file.

350
360
370 } Access files and display messages.
380

390 Close both files.
400
410 End program.

Sample Program Output

```
RUN
ENTER DATE AND TIME (DD-MOM-YY HH:MM:SS):12-DEC-77 08:30:30
HOW MANY DEVICES ON YOUR SYSTEM?:1
ENTER DEVICE ADDRESSES:0


THIS IS A SAMPLE PROGRAM FOR DEVICE 0

FORMAT REQUESTED, OK TO DESTROY DATA ON DEVICE  0?Y
THIS IS AN ASCII SAMPLE (SEQUENTIAL FILE)
THIS IS A BINARY SAMPLE (RANDOM FILE)
```

HOW TO WRITE A FILE IDENTIFIER

The File Identifier (F.I.) is placed in commands to tell the system where it can find an existing file or where to create a new file. The F.I. also assigns names to up to four preceding libraries and to the file itself. It is also used to assign passwords and the extension.

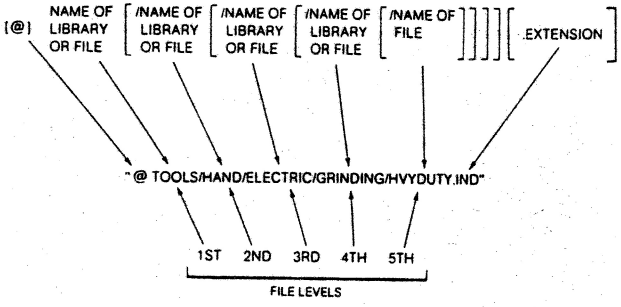
An F.I. can be as small as this example representing a file on the 1st level:

"TOM"

An F.I. can be as large as this one representing five levels, each with its own password:

"FEDERAL:RED/STATE:BLUE/COUNTY:GREEN/MUNICIPAL:
YELLOW/LOCAL:BLACK"

The descriptive format for an F.I. and how it is used to write a typical F.I. is shown below.


[@] NAME OF LIBRARY OR FILE [/NAME OF LIBRARY OR FILE [/NAME OF LIBRARY OR FILE [/NAME OF LIBRARY OR FILE [/NAME OF FILE]]]] [.EXTENSION]
"@ TOOLS/HAND/ELECTRIC/GRINDING/HVYDUTY.IND"
1ST 2ND 3RD 4TH 5TH
FILE LEVELS

FIELD DEFINITIONS

@	The commercial "at" sign is necessary anytime you wish to avoid accessing the current library. In this example the system will go directly to the library TOOLS.
TOOLS:	This is the name of the 1st level library.
HAND:	This is the name of the 2nd level library.
ELECTRIC:	This is the name of the 3rd level library.
GRINDING:	This is the name of the 4th level library.
HVYDUTY:	This is the name of the 5th level file.

.IND: This is the extension allowing the system to distinguish between this and some other HVYDUTY file under these same libraries. Only a file may have an extension.

Up to 21 characters of the F.I. may be placed in a CALL "USERLIB" command. These characters would then be considered the "current library" and would not need to be repeated in subsequent commands.

PASSWORDS

A password may be added to any file or library name to prevent unauthorized access, for example:

TOOLS:PASSWRD

This may be done only when the file or library is created.

DELIMITERS

A slash (/) separates name fields:

TOM/FRED

The entire F.I. must be contained in quotation marks:

"TOM/FRED"

A colon (:) separates the password from the name of the library or file:

"TOM:PASSWRD/FRED"

A period (.) separates the extension from the file name or its password:

"TOM:PASSWRD/FRED.EXT"

SAMPLE DEVICE AND FILE STATUS MESSAGES

Sample File Status Message (full)

```
BRS N ATR      30208 ALLOC      12-DEC-77  08:30 ALT
               10000 USED       12-DEC-77  10:30 USED
    1 OPEN       200 REC LEN    12-DEC-77  08:30 CREATED
UNIVERSITY/RECORD
```

This sample shows that a file with the F.I. UNIVERSITY/RECORD was created December 12, 1977 at 8:30 AM and last used December 12, 1977, at 10:30 AM. This message shows the file has never been altered because the ALT field is exactly the same as the CREATED field. The file's attributes (BRSN) show the file is binary, private, scattered and non-compressible. Approximately 30K bytes of space have been allocated to this file in record lengths of 200 bytes, and 10K bytes have already been used.

Sample Device Status Message

```
00 UNIT
4907          DEV     ID          INCTAX          VOL          ID
SMITH & CO.    OWNER      250000                 630000 SIZE
2560   LOST   256 BLK SIZE   07-MAR-77 10:30 FORMATTED
00003  OPEN
```

This message would be generated for a device identified as the 4907 and as 00 UNIT[2]. The disc is identified as INCTAX and the owner as SMITH & CO. There are 250,000 bytes of usable non-file space. The 630,000-size field indicates this is a dual density flexible disc with 2,560 bytes of unusable or damaged space. The block size is 256.

The disc was last formatted on March 7, 1977, at 10:30 AM. Three files on the disc are open and the file is not write-protected.

File status messages on open files follow the device status messages.

[2] Unit number appears only when CALL "CUSTAT" is executed.

NOTES